Robust Super-Resolution of Real Faces Using Smooth Features

Saurabh Goswami(✉), Aakanksha, and A. N. Rajagopalan

Indian Institute of Technology, Madras, India
{ee18s003,ee18d405}@smail.iitm.ac.in, raju@ee.iitm.ac.in

Abstract. Real low-resolution (LR) face images contain degradations which are too varied and complex to be captured by known downsampling kernels and signal-independent noises. So, in order to successfully super-resolve real faces, a method needs to be robust to a wide range of noise, blur, compression artifacts etc. Some of the recent works attempt to model these degradations from a dataset of real images using a Generative Adversarial Network (GAN). They generate synthetically degraded LR images and use them with corresponding real high-resolution (HR) image to train a super-resolution (SR) network using a combination of a pixel-wise loss and an adversarial loss. In this paper, we propose a two module super-resolution network where the feature extractor module extracts robust features from the LR image, and the SR module generates an HR estimate using only these robust features. We train a degradation GAN to convert bicubically downsampled clean images to real degraded images, and interpolate between the obtained degraded LR image and its clean LR counterpart. This interpolated LR image is then used along with it's corresponding HR counterpart to train the super-resolution network from end to end. Entropy Regularized Wasserstein Divergence is used to force the encoded features learnt from the clean and degraded images to closely resemble those extracted from the interpolated image to ensure robustness.

1 Introduction

Face Super-Resolution (SR) is an important preprocessing step for high-level vision tasks like facial detection and recognition. Robustness to real degradations like noise, blur, compression artifacts, etc. is one of the key aspects of the human visual system and hence highly desirable in machine vision applications as well. Incorporating this robustness in the Super-Resolution stage itself would ease all the downstream tasks. Unfortunately, most of the face SR methods are trained with a fixed degradation model (downsampling with a known kernel and adding noise) that is unable to capture the complexity and diversity of real degradations

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-66415-2_11) contains supplementary material, which is available to authorized users.

A. Bartoli and A. Fusiello (Eds.): ECCV 2020 Workshops, LNCS 12535, pp. 169–185, 2020.
https://doi.org/10.1007/978-3-030-66415-2_11

and hence performs poorly when applied on real degraded face images. This problem becomes more pronounced when the image is extremely small. Since most of the useful information is degraded, it further increases the ambiguity in reconstruction process. Previous methods such as [3,32,34] use facial heatmaps and facial landmarks as priors to reduce ambiguity. [30,33] leverage autoencoders to build networks which are robust to synthetic noise and [18] leverage wavelet transform to train a network which is robust to gaussian noise. However, none of the above methods have been proven to be robust to real degradation except [3]. In [4], a Generative Adversarial Network (GAN) was trained to generate realistically degraded Low-Resolution (LR) versions of clean High-Resolution (HR) face images and another GAN was trained to super-resolve the synthetic degraded images to their corresponding clean HR counterparts. To the best of our knowledge, this is the only previous work which super-resolves real degraded faces without the aid of any facial priors. However, we observed that [4] produces visually different outputs for different degradations. This can be attributed to the fact that the network sees every degraded image independently and there is no explicit constraint to extract the same features from different degraded versions of the same image.

In this paper, we focus on incorporating robustness to degradations in the task of tiny face super-resolution without the need of a face specific prior and without a dataset of *degraded LR-clean HR* image pairs. Premised upon the observation that humans are remarkably adept at registering different degrdaded versions of the same image as visually similar images, we prepend a smooth feature extractor module to our Super-Resolution (SR) module. Since our feature extractor is smooth with respect to real degradations, its output does not vary wildly when we move from clean images to degraded images. The SR module which produces clean HR images from features extracted by the smooth feature extractor, thus, produce similar images regardless of the degradation. Features which remain smooth under degradations are also features that are common between clean and degraded LR. So, our network, in essence, learns to look at features which are similar between clean and degraded LR.

Following [4], we train a GAN to convert clean LR images to corresponding degraded LR images. One training iteration of our network involves two backpropagations. During the first backpropagation, we update parameters of both modules of our network to learn a super-resolution mapping from an interpolated LR (by combining clean and degraded LR) to its corresponding clean HR. The interpolation is carried out to avoid having the network overfit one of two LR domains (clean and degraded). During the second backpropagation, we minimize the Entropy Regularized Wasserstein Distance between features extracted from clean as well as degraded LR and those extracted from interpolated LR. The interpolation also helps in ensuring smoothness of the feature extractor.

During test time, we put an image (clean or degraded) through the feature extractor module first and then feed the extracted features to the SR Module to get the corresponding super-resolved image. Since the extracted features do not change significantly between clean and degraded images, the super-resolution output for a degraded image does not change significantly from that of a clean image. We perform tests to visualise the robustness of our network as well as smoothness of the features extracted by our feature extractor.

The main contributions of our work are as follows:

- We propose a new approach for unpaired face SR where the SR network relies on features that are common between corresponding clean and degraded images.
- To the best of our knowledge, ours is the first work that handles robustness separately from the task of super-resolution. This enables us to explicitly enforce robustness constraints on the network.

2 Related Works

Single Image Super-Resolution (SISR) is a highly ill-posed inverse problem. Traditional methods mostly impose handcrafted constraints as priors to restrict the space of solutions. With the availability of Large-scale Image Datasets and the consistent success of Convolutional Neural Networks (CNNs), learning (rather than handcrafting) a prior from a set of natural images became a possibility. Many such approaches have been explored subsequently.

2.1 Deep Single Image Super-Resolution

We classify all the deep Single Image Super-Resolution (SISR) methods in two broad categories - (i) deep Paired SISR and (ii) deep Unpaired SISR. In paired SISR, corresponding pairs of LR and HR images are available and the network is evaluated on its ability to estimate an HR image given its LR counterpart. Most of the available deep paired SISR networks are trained under a setting where LR images are generated by downsampling HR images (from datasets such as Set5, Set14, DIV2K [1], BSD100 [2] etc.) using a known kernel (often bicubic). These networks are trained using either a pixel wise Mean Squared Error (MSE) loss e.g. [13,21,26], L_1 loss e.g. [36], Charbonnier loss e.g. [22] or a combination of pixel-wise L_1 loss, perceptual loss [20] and adversarial loss [16] e.g. [10,23,29]. Even though these networks perform really well in terms of PSNR and SSIM, and the GAN based ones produce images that are highly realistic, these networks often fail when they are applied on real images with unseen degradations such as realistic noise and blur. To address this, RealSR [6] dataset was introduced in NTIRE 2019 Challenge [5] containing images taken at two different focal lengths of a camera. Networks like [14,15,19] were trained on this dataset and are therefore robust to real degradations.

On the other hand, in unpaired SISR, only the LR images are available in the dataset. In [35], a CycleGAN [37] was trained to denoise the input image and another one to finetune a pretrained super-resolution network. In [27], a CycleGAN was trained to generate degraded versions of clean images and a super-resolution network was then trained using pairs of synthetically degraded

LR and clean HR images.
However, all these networks are meant for natural scenes and not faces in particular. Humans are highly sensitive to even the subtlest changes when it comes to human faces, making the task of perceptually super-resolving human faces a challenging and interesting one.

2.2 Deep Face SISR

General SR networks as the ones mentioned above, often produce undesired artifacts when applied on faces. Hence, paired face SR networks often rely on face-specific prior information to subdue the artifacts and make the network focus on important features.

Networks like [3,10,32,34] rely on facial landmarks and heatmaps to impose additional constraints on the output whereas [12] leverage HR exemplars to produce high-quality HR outputs. On the other hand, networks like [9,18] rely on pairs of LR and HR face images to perceptually super-resolve faces. Even though the above methods are somewhat robust to noise and occlusion, they are not equipped well enough to handle noises which are as complex and as diverse as those in real images. [30,33] leverage capsule networks and transformative autoencoders to class-specifically super-resolve noisy faces but the noises are synthetic. As of yet, there seems to be no dataset with paired examples of degraded LR and clean HR images of faces available. As a result, in recent years, there has been a shift in face SISR methods from paired to unpaired. Recently, with the release of *Widerface* [31] dataset of real low-resolution faces and the wide availability of high resolution face recognition datasets such *AFLW* [28], *VGGFace2* [7] and *CelebAMask-HQ* [24], Bulat et al. [4] propose a training strategy where a High-to-Low GAN is trained to convert instances from clean HR face images to corresponding degraded LR images and a Low-to-High GAN is then trained using synthetically degraded LR images and their clean HR counterparts. This method is highly effective since it does not require facial landmarks or heatmaps for faces (as they are not available for real face images captured in the wild).

However, despite producing sharp outputs, it is not very robust as different outputs are obtained for different degradations in the LR images. In order to explicitly impose robustness, we introduce a smooth feature extractor module to extract similar features from a degraded LR image and its clean LR counterpart. This enabled us to get features that are more representative of the actual face in the image and is significantly less affected by the degradations in the input.

2.3 Robust Feature Learning

Our work builds on the existing methods in robust feature learning. Haoliang et al. [25] extract robust features from multiple datasets of similar semantic contents by minimizing Maximum Mean Discrepancy (MMD) between features extracted from these datasets. Cemgil et al. [8], achieve robustness by forcing Entropy Regularized Wasserstein Distance to be low between features extracted from clean images and their noisy counterparts. None of these works handle

Super-Resolution where rigorous compression using an autoencoder may hurt the reconstruction quality. We propose a method of incorporating robust feature learning in super-resolution without requiring any face specific prior information.

3 Proposed Method

3.1 Motivation

Super-Resolution networks which are meant to be used on real facial images need to satisfy two criteria: (i) they need to be robust under real degradations, (ii) they should preserve the identity and pose of a face. Deep state-of-the-art super-resolution networks usually derive the LR images by bicubically downsampling HR images. Hence, an SR network trained on pairs of LR and HR images used for training fail to meet the first criterion. On the other hand, SR networks trained with real degradations fail to satisfy the second criterion. Noting the fact that the face recognition ability of us humans does not change very significantly with reasonably high degradation in images, it should be possible to find features that remain invariant under significant degradation and train a super-resolution network that would rely only on these features. Now, features which are robust to degradations would also be smooth under the said degradations. So, by enforcing explicit smoothness constraints on the extracted features, we can ensure robustness.

3.2 Overall Pipeline

We have a clean High-Resolution dataset Y_c and a degraded Low-Resolution dataset X_d. We obtain clean Low-Resolution dataset, X_c, corresponding to Y_c, by downsampling every image in Y_c with a bicubic downsampling kernel. So every x_c in X_c is a downsampled version of some y_c in Y_c, using the equation

$$x_c = (y_c * k)_{\downarrow s} \tag{1}$$

where, k is the bicubic downsampling kernel and s is the scale factor. Following [4], we train a Degradation GAN, G_d to convert clean samples from X_c to look like they have been drawn from the degraded LR dataset X_d. We call this synthetic degraded LR dataset $\widehat{X_d}$ and samples in this dataset $\widehat{x_d}$. So,

$$\widehat{x_d} = G_d(x_c, z) \in \widehat{X_d} \quad \forall \quad x_c \in X_c \tag{2}$$

where $z \in Z$ is an additional vector input which is sampled from a distribution Z to capture the one-to-many relation between HR and degraded LR images.

Our network basically comprises 2 modules - (i) Feature Extractor Module (f) and (ii) Super-Resolution Module (g). During training, we first sample an x_c from X_c and generate one of its degraded counterparts $\widehat{x_d} = G_d(x_c, z)$ using G_d. We then combine these two LR images with a mixing coefficient α

$$x_{in} = \alpha x_c + (1 - \alpha)\widehat{x_d} \tag{3}$$

where $0 < \alpha < 1$. We, then, put x_{in} through the convolutional feature extractor $f(x)$ and the SR module $g(h)$ to estimate the corresponding clean HR output $\widehat{y_c}$ and do a backpropagation.

$$h_{in} = f(x_{in}), \quad \widehat{y_c} = g(h_{in}) \tag{4}$$

To ensure smoothness of f under real degradations, we extract features h_c and h_d from x_c and $\widehat{x_d}$

$$h_c = f(x_c) \quad h_d = f(\widehat{x_d}) \tag{5}$$

and minimize the Entropy Regularized Wasserstein Distance (Sinkhorn distance) between (h_c, h_{in}) and (h_d, h_{in}) through another backpropagation. We recalculate h_{in} during this operation as well. Figure 1 shows a schematic diagram of our approach.

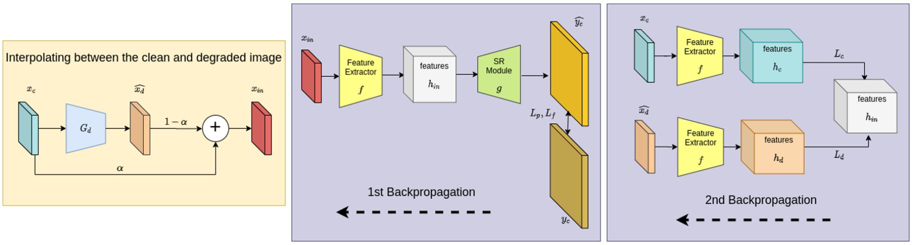


Fig. 1. The proposed approach.

Here, if we use $\alpha = 0$, since the entire network, during the first backpropagation, would be trained using pairs of synthetically degraded LR and clean HR samples, it may end up learning a mapping that would fail to preserve the identity of a face. However, if we take $\alpha = 1$, the network may exhibit preference to the domain of clean LR images. So, we needed an input LR image which is not as sharp as x_c but not as degraded as $\widehat{x_d}$ either. Since the edges in x_c are much sharper than those in $\widehat{x_d}$, x_{in} continues to appear reasonably clean even when $\alpha < 0.5$. This is why we do not sample α from a distribution since that might end up giving one domain advantage over the other and keep it fixed at 0.3 since $\alpha = 0.3$ appears to us to have struck the right balance between the two LR domains visually.

Also, using $0 < \alpha < 1$, enables us to apply the smoothness constraint between (h_c, h_{in}) and (h_d, h_{in}) which is a better way to ensure smoothness than imposing smoothness constraint on pairs of (h_c, h_d).

3.3 Modeling Degradations with Degradation GAN

Owing to the complex and diverse nature of real degradations, it is extremely difficult to mathematically model them by hand. So, following previous works [4,27], we train a GAN (termed Degradation GAN) to model real degradations.

Generator. Our Degradation GAN Generator, shown in Fig. 3, G_d, has 3 downsampling blocks, each consisting of a ResNet block followed by a 3×3 convolutional with *stride* $= 2$, and 3 upsampling blocks each comprising ResNet blocks followed a Nearest Neighbour Upsampling layer and a 3×3 convolutional block with *stride* $= 1$. The downsampling and upsampling paths are connected through skip connections. All the ResNet blocks used in Generator follow the structure described in Fig. 2. Our Generator takes a bicubic downsampled image x_c and an n dimensional random vector z sampled from a normal distribution. We expand each of the n dimensions of the random vector into a channel of size $H \times W$ (filled with a single value) where H and W are the height and width of every image. We concatenate the expanded volume with the image and feed it to the generator.

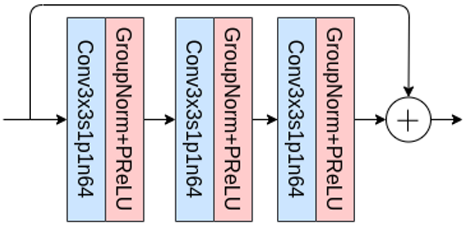


Fig. 2. ResNet block used in Degradation GAN Generator.

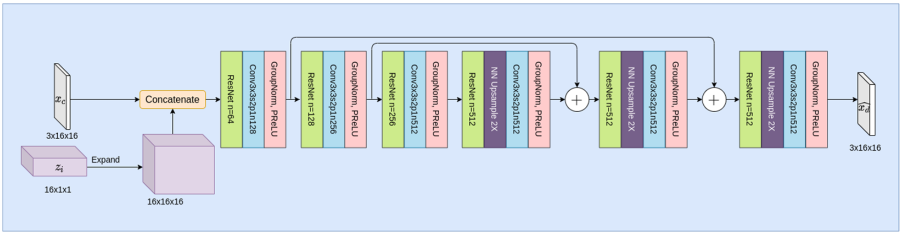


Fig. 3. The overall architecture of the Degradation GAN Generator G_d.

Critic. We use the same discriminator used in [23]. Since we train the degradation GAN as Wasserstein GAN [17], we replace the Batch Normalization layers with Group Normalization and remove the last Sigmoid layer. Following the nomenclature, we call it critic instead of discriminator.

Loss Functions. We train the degradation GAN as a Wasserstein GAN with Gradient Penalty (WGAN-GP) [17]. So, the critic is trained by minimizing the following loss function:

$$L_D = (\mathbb{E}_{x \in \widehat{X_d}}[D(x)] - \mathbb{E}_{x \in X_d}[D(x)]) + \lambda \mathbb{E}_{\widehat{x} \sim \mathbb{P}_{\widehat{x}}}[(\|\nabla_{\widehat{x}} D(x)\|_2 - 1)^2] \quad (6)$$

where, as in [17], the first term is the original critic loss and the second term is the gradient-penalty.

To maintain the correspondence between inputs and outputs of the generator, we add a Mean Square Loss (MSE loss) term to the WGAN loss in the objective function L_G of the generator.:

$$L_G = \lambda_{WGAN} L_{WGAN} + \lambda_{MSE} L_{MSE} \tag{7}$$

where,

$$L_{WGAN} = -\mathbb{E}_{(x_c,z)\in(X_c,Z)}[G_d(x_c,z)] \tag{8}$$

and

$$L_{MSE} = \|x_c - G_d(x_c,z)\|^2 \tag{9}$$

3.4 Super-Resolution Using Smooth Features

The main objective of our work is to design a robust SR network the performance of which does not deteriorate under real degradation. Our network has two modules (a) a fully-convolutional feature extractor f and (b) a fully-convolutional SR module g. The way we achieve robustness is by making the feature extractor smooth under degradations and making the SR module g rely solely on the features extracted by f. In [8], Cemgil et al. proposed a method to enforce robustness on the representations learnt by Variational Autoencoders (VAEs). They trained a VAE to reconstruct clean images and minimized the Entropy Regularized Wasserstein Distance between representations derived from a clean image and its noisy version.

There were three challenges in applying this method to Super-Resolution:

1. Autoencoders compress an input down to its most important components and ignore information like occlusion, background objects, etc. For accurate reconstruction of an HR image from its LR counterpart, it is important to preserve this information. Hence, we cannot perform a rigorous dimensionality reduction. On the other hand, if we decide to keep the dimensionality intact, it will make it harder to achieve robustness since there are too many distractors. So it is important to choose a reduction factor that will achieve the best trade-off between reconstruction and robustness.
2. They train their network for synthetic noise. However, real degradation involves signal-dependent noise, blur and a variety of other artifacts. So, we need a mechanism to realistically degrade images.
3. As we show in the supplementary material, despite smoothness constraint and despite the network being reasonably robust, naively applying their method on SR still leaves a gap between its performance on clean and degraded images. So, we need a better training strategy.

To address (1), we try a number of different dimensionality reduction choices $(1\times, 4\times, 16\times)$ for the features extracted by f and we observed that $4\times$ dimensionality reduction attains the best trade-off. To address (2), we train a degradation GAN to realistically degrade clean images. To address (3), we interpolate between a clean image (x_c) and one of its synthetically degraded counterpart ($G_d(x_c, z)$) using a mixing coefficient α as shown in Eq. 3. We call this x_{in}.

Feature Extractor f: Our feature extractor consists of 4 Residual Channel Attention (RCA) downsampling and 2 upsampling blocks. As shown in Fig. 4, there are 2 skip connections. It is a fully convolutional module which takes an LR image of dimension $3 \times 16 \times 16$ at the input and produces a feature volume of dimension $64 \times 4 \times 4$. In Fig. 4, 'RCA, n64' denotes an RCA block with 64 output channels and 'Conv3x3, s2 p1 n64' denotes a 3×3 convolutional layer with $stride = 2$, $padding = 1$ and 64 output channels.

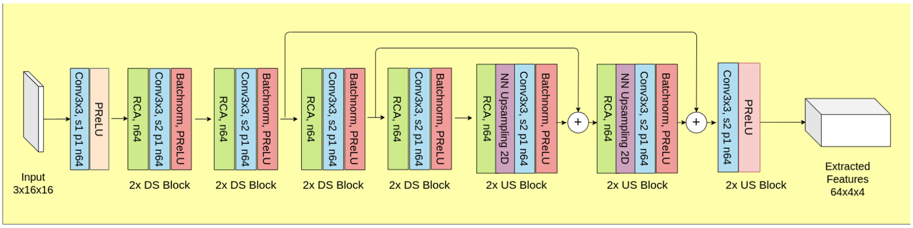


Fig. 4. Feature extractor f.

Super-Resolution Module g: Our Super-Resolution module consists of 6 upsampling blocks and 2 DenseBlocks as shown in Fig. 5. The upsampling blocks comprise a Pixel-Shuffle layer, a convolution layer, a Batch-Normalization layer and a PReLU layer. The DenseBlocks contain a number of Residual Channel Attention (RCA) blocks and Residual Channel Attention Back-Projection (RCABP) blocks connected in a dense fashion as in [19]. In Fig. 5, 'Pixel Shuffle (2)' denotes 2x pixel-shuffle upsampling layer and 'RCABP, n64' stands for an RCABP block with 64 heatmaps at the output.

During one forward pass, we pass a minibatch of x_{in} through our feature extractor f to produce the feature volume h_{in}. We put h_{in} through our Super-Resolution module g to produce a high resolution estimate $\widehat{y_c}$ and do a back propagation through both g and f. This ensures that the features are useful for SR. Since x_{in} is neither as clean as x_c nor as severely degraded as $\widehat{x_d}$, the possibility of our SR network being biased to any one of the domains is eliminated.

After the first backpropagation, we put one minibatch each of $x_c, \widehat{x_d}$ and x_{in} (again) through f, as shown in Eq. 5, and calculate the Sinkhorn Distance [11] (which calculates the Entropy Regularized Wasserstein Divergence) between (h_c, h_{in}) and (h_d, h_{in}),

$$L_c = Sinkhorn(h_c, h_{in}), \quad L_d = Sinkhorn(h_d, h_{in}) \tag{10}$$

Using a combination of L_c and L_d as a loss function, we backpropagate through f one more time to enforce smoothness under degradations.

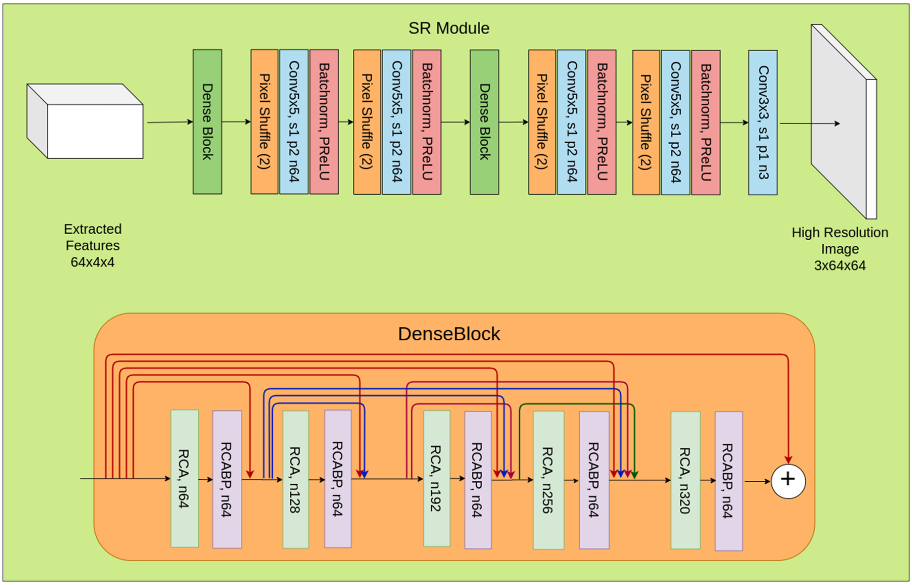


Fig. 5. Architecture of SR module g and DenseBlock.

Like our Degradation GAN, we train our robust super-resolution network (during the first back propagation) like a Wasserstein GAN. So, the objective function here is a combination of adversarial loss (L_{adv}), pixel-level L_1 loss (L_p) and a perceptual loss [20] (L_f) computed between features extracted from the estimated ($\widehat{y_c}$) and ground-truth (y_c) HR images through a subset of VGG16 network. Hence, the overall objective function optimized during the first back propagation is

$$L_{sr} = \lambda_p L_p + \lambda_f L_f + \lambda_{adv} L_{adv} \tag{11}$$

where,

$$L_p = \|y_c - \widehat{y_c}\|_1 \tag{12}$$

$$L_f = \|f_{vgg}(y_c) - f_{vgg}(\widehat{y_c})\|_1 \tag{13}$$

$$L_{adv} = -\mathbb{E}_{x_{in}\sim\widehat{\mathbb{P}_x}}[D_{sr}(g(f(x_{in})))] \tag{14}$$

with f_{vgg} being a subset of VGG16 network, $\mathbb{P}_x$ being the distribution described by x_{in} and D_{sr} being the critic comparing the generated HR images with the ground-truth HR images. The architecture of D_{sr} is same as the critic of degradation GAN and it is trained with the following loss function:

$$L_{DSR} = (\mathbb{E}_{\widehat{y_c}\sim\mathbb{P}_y}[D_{sr}(\widehat{y_c})] - \mathbb{E}_{y_c\in Y_c}[D(y_c)]) + \lambda\mathbb{E}_{\widehat{y}\sim\mathbb{P}_{\widehat{y}}}[(\|\nabla_{\widehat{y}}D(\widehat{y})\|_2 - 1)^2] \tag{15}$$

where $\mathbb{P}_y$ is the distribution generated by the outputs of our network and $\mathbb{P}_{\widehat{y}}$ is the distribution of samples interpolated between $\widehat{y_c}$ and y_c.

For the second back propagation, we optimize a combination of the Sinkhorn Distances mentioned earlier

$$L_{robust} = \lambda_c L_c + \lambda_d L_d \tag{16}$$

Since the second backpropagation is only through f, it does not directly affect the mapping learnt by g and only makes f smooth under degradations.

4 Experiments

4.1 Training Details

We use two-time step update for both our Degradation GAN and Robust Super-Resolution Network. For both D and D_{sr}, we start with a learning rate of 4×10^{-4} and decrease them by a factor of 0.5 after every 10000 iterations. For all the other networks (G_d, f, g) we set the initial training at 10^{-4} and decay it by a factor of 0.5 after every 10000 iterations.

For all networks, we use Adam Optimizer with $\beta_1 = 0.0$ and $\beta_2 = 0.9$. For every 5 updates of discriminators, we update the corresponding generator networks once. We try out a number of different values of λ and the ones that worked best for us are $[\lambda_{WGAN} = 0.05, \lambda_{MSE} = 1, \lambda_p = 1, \lambda_f = 0.5, \lambda_{adv} = 0.05, \lambda_c = 0.3, \lambda_d = 0.7]$. For G_d, we sample z from a 16−dimensional multivariate normal distribution with zero mean and unit standard deviation.

4.2 Datasets

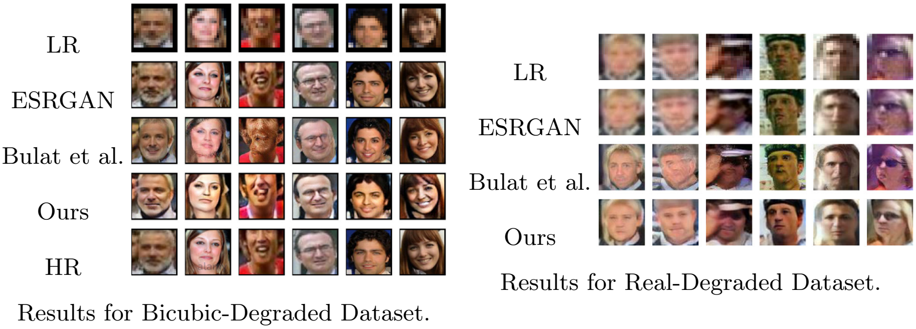
We train our network for 4× super-resolution ($s = 4$). However, our robustness strategy is not scale dependent. For training our network, we used two datasets: one with degraded images and the other with clean images. To make the degraded image dataset, we randomly sample 153446 images from the *Widerface* [31] dataset. This dataset contains face images with a wide range of degradations such as: varying degrees of noise, extreme poses and expressions, occlusions, skew, non-uniform blur etc. We use 138446 of these images for training and 15000 for testing. While compiling the clean dataset, to make sure it is diverse enough in terms of poses, occlusions, skin colours and expressions, we combined the entire *AFLW* [28] dataset with 60000 images from *CelebAMask-HQ* [24] dataset and 100000 images from *VGGFace2* [7] dataset. To obtain clean LR images, we simply downsample images from the clean dataset.

4.3 Results

To assess the accuracy as well as robustness of our work, we test our network on 3 different datasets - (i) Bicubically-Degraded Dataset, (ii) Synthetically-Degraded Dataset and (iii) Real-Degraded Dataset.

1. **Bicubically-Degraded Dataset:** To compile this dataset, we randomly sample 4000 HR images from the clean Facial Recognition Datasets as mentioned above. We bicubically downsample them to obtain paired LR-HR images. Evaluation on this dataset tells us about the reconstruction accuracy of our SR network.

   As shown in Fig. 6a, ESRGAN [29] performs best on this dataset since it was trained on bicubic downsampled images. Interestingly, the results of [4] appear to be a little different from the HR ground truth in terms of identity. We observe this in all our experiments. Also, their outputs contain a lot of undesired artifacts. Our outputs are faithful to the ground-truth HR and contain less artifacts. However, our method performs a little poorly in terms of PSNR and SSIM as shown in Table 1. However, since we focus primarily on the robustness part of the problem, the strength of our network becomes evident with the evaluation of robustness.



Results for Bicubic-Degraded Dataset.

Results for Real-Degraded Dataset.

Fig. 6. Comparison of results.

Table 1. Comparison of PSNR/SSIM on Bicubic-Degraded Dataset

Method	PSNR	SSIM
ESRGAN [29]	**25.25**	**0.351**
Bulat et al. [4]	24.55	0.220
Ours	24.48	0.218

2. **Synthetically-Degraded Dataset:** This dataset contains the same HR images as the Bicubically-Degraded Dataset but we obtain 5 different synthetically degraded LR versions of each HR image using our degradation GAN. We perform two tests on this dataset:
   - **Robustness Test:** Here, for each HR image, we put all 5 degraded LR images through our SR network. This test shows us how the output changes for different degradation. The similarity between the outputs will tell us how robust our network is to realistic degradations which is the focus of our work.

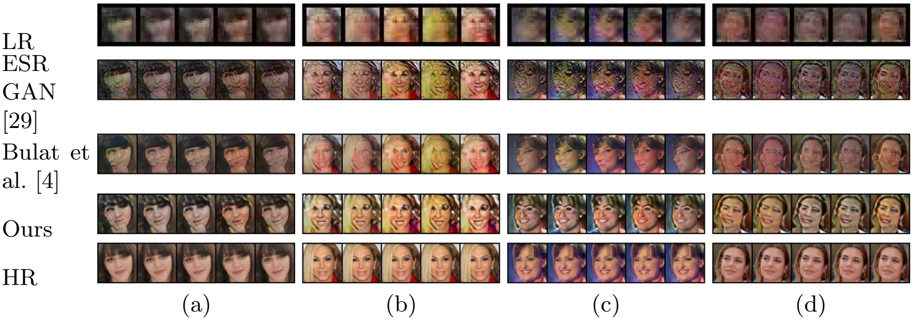


Fig. 7. Comparison of robustness.

As shown in Fig. 7, these images are extremely degraded. ESRGAN [29] gives the worst performance on this dataset. [4] produces slightly different-looking faces for different degradations. Our method, however, produces outputs that look similar for all these degradations. This shows that our network is robust to realistic degradations.

- **Smoothness Test:** This experiment enables us to visualise the smoothness of our feature extractor (f). Here, we combine every degraded LR image in the dataset with their bicubically downsampled counterparts using 5 different values of α_i such as $[0.0, 0.2, 0.4, 0.8, 1.0]$ to create a set of 5 different images ($\{x_{mix}\}$). Since α is the coefficient we use to mix clean and corresponding degraded images, by gradually varing α from 0 to 1 and noting the output, we get an idea of how adept our network is at maintaining its output as we gradually move from a clean image, through increasingly degraded images, to one of its realistically degraded versions. If our network manages to maintain its output without altering its overall appearance (changing pose, identity, etc.), it would mean that the learnt features are smooth and robust to degradations.

$$x_{mix}^{i} = \alpha_i x_c + (1 - \alpha_i)\widehat{x_d} \tag{17}$$

Figure 8 shows a comparison of the output of our network with those of [4] and [29]. The outputs of ESRGAN [29] becomes increasingly worse as α decreases. The outputs of [4] changes significantly as α goes from 0 to 1, sometimes even producing different faces. The output of our network does not undergo any visually significant changes. This establishes the features learnt by the feature extractor are smooth under realistic degradation. Figure 7 shows that ESRGAN [29] consistently performs poorly in terms of robustness than the other two methods. This is expected since it was trained with bicubically downsampled LR images only. The behavior of [4] is interesting. In Fig. 7(a), (b) and (g), it is generating additional facial components that are unrelated to the content of the input. The performance of our network, as shown in (f) and (g), drops a little when the input is heavily degraded but the recognizable features do not change much.

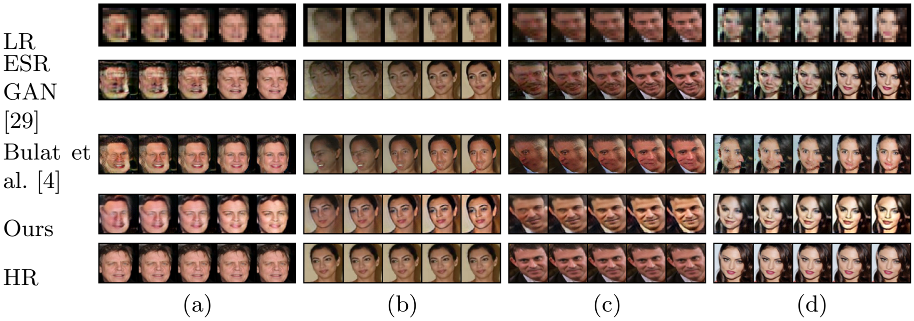


Fig. 8. Visualizing smoothness for $\alpha = [0, 0.2, 0.4, 0.8, 1.0]$.

3. **Real-Degraded Dataset:** This dataset contains 15000 images from the *Widerface* Dataset. Performance on this dataset will dictate how effective our method is in super-resolving real degraded facial images.
   As shown in Fig. 6b, our method is able to super-resolve real degraded faces. The outputs of [4] contain undesired artifacts and sometimes exhibit identity discrepancy as well. ESRGAN [29] is able to maintain the identity but the outputs are not sharp. Since we do not have ground-truth HR images for these LR images, we can not compute PSNR/SSIM. So, we use Fretchet Inception Distance (FID) as a metric to assess how close the output is to the target distribution of sharp images. Table 2 shows the FIDs of [4,29] and our method computed over 15000 images. Lower FID denotes better adherence to target distribution and hence sharper output.

Table 2. Comparison of FID.

Method	FID
ESRGAN [29]	139.2599
Bulat et al. [4]	**74.2798**
Ours	77.1359

   As shown in Table 2, our method performs very close to [4] in terms of realness of the output and at the same time, maintains a fixed output under varying degradations. So, our method is robust and at the same time, effective on real degraded faces.

5 Conclusion

We propose a robust super-resolution network that would give consistent output under a wide range of degradations. We train a feature extractor that is able to extract similar features from both bicubically downsampled images and their corresponding realistically degraded counterparts. We perform robustness test to put our claim of robustness to test and smoothness test to visualize the variation in extracted features as we gradually move from a clean to a degraded LR image. There is still room to improve our network for better performance in terms of PSNR/SSIM. In our future works, we will attempt to address this.

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: dataset and study. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, July 2017
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 898–916 (2011). https://doi.org/10.1109/TPAMI.2010.161
3. Bulat, A., Tzimiropoulos, G.: Super-fan: integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with GANs. CoRR abs/1712.02765 (2017). http://arxiv.org/abs/1712.02765
4. Bulat, A., Yang, J., Tzimiropoulos, G.: To learn image super-resolution, use a GAN to learn how to do image degradation first. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11210, pp. 187–202. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01231-1_12
5. Cai, J., Gu, S., Timofte, R., Zhang, L.: Ntire 2019 challenge on real image super-resolution: methods and results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2019)
6. Cai, J., Zeng, H., Yong, H., Cao, Z., Zhang, L.: Toward real-world single image super-resolution: a new benchmark and a new model. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
7. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: a dataset for recognising faces across pose and age. In: International Conference on Automatic Face and Gesture Recognition (2018)
8. Cemgil, T., Ghaisas, S., Dvijotham, K.D., Kohli, P.: Adversarially robust representations with smooth encoders. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=H1gfFaEYDS
9. Chen, X., Wang, X., Lu, Y., Li, W., Wang, Z., Huang, Z.: RBPNET: an asymptotic residual back-projection network for super-resolution of very low-resolution face image. Neurocomputing **376**, 119–127 (2020). https://doi.org/10.1016/j.neucom.2019.09.079. http://www.sciencedirect.com/science/article/pii/S0925231219313530
10. Chen, Y., Tai, Y., Liu, X., Shen, C., Yang, J.: FSRNet: end-to-end learning face super-resolution with facial priors. CoRR abs/1711.10703 (2017). http://arxiv.org/abs/1711.10703
11. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transportation distances (2013)
12. Dogan, B., Gu, S., Timofte, R.: Exemplar guided face image super-resolution without facial landmarks. CoRR abs/1906.07078 (2019). http://arxiv.org/abs/1906.07078
13. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. CoRR abs/1501.00092 (2015). http://arxiv.org/abs/1501.00092

14. Du, C., Zewei, H., Anshun, S., Jiangxin, Y., Yanlong, C., Yanpeng, C., Siliang, T., Ying Yang, M.: Orientation-aware deep neural network for real image super-resolution. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019
15. Feng, R., Gu, J., Qiao, Y., Dong, C.: Suppressing model overfitting for image super-resolution networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019
16. Goodfellow, I.J., et al.: Generative adversarial networks (2014)
17. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. CoRR abs/1704.00028 (2017). http://arxiv.org/abs/1704.00028
18. Huang, H., He, R., Sun, Z., Tan, T.: Wavelet-SRNET: a wavelet-based CNN for multi-scale face super resolution. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1698–1706 (2017)
19. Jang, D., Park, R.: DenseNet with deep residual channel-attention blocks for single image super resolution. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1795–1803 (2019)
20. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution (2016)
21. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. CoRR abs/1511.04587 (2015). http://arxiv.org/abs/1511.04587
22. Lai, W., Huang, J., Ahuja, N., Yang, M.: Fast and accurate image super-resolution with deep Laplacian pyramid networks. CoRR abs/1710.01992 (2017). http://arxiv.org/abs/1710.01992
23. Ledig, C., et al.: Photo-realistic single image super-resolution using a generative adversarial network. CoRR abs/1609.04802 (2016). http://arxiv.org/abs/1609.04802
24. Lee, C.H., Liu, Z., Wu, L., Luo, P.: MaskGAN: towards diverse and interactive facial image manipulation. arXiv preprint arXiv:1907.11922 (2019)
25. Li, H., Jialin Pan, S., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
26. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. CoRR abs/1707.02921 (2017). http://arxiv.org/abs/1707.02921
27. Lugmayr, A., Danelljan, M., Timofte, R.: Unsupervised learning for real-world super-resolution (2019)
28. Martin Koestinger, Paul Wohlhart, P.M.R., Bischof, H.: Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization. In: proceedings of the First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies (2011)
29. Wang, X., et al.: ESRGAN: enhanced super-resolution generative adversarial networks. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11133, pp. 63–79. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11021-5_5
30. Xin, J., Wang, N., Jiang, X., Li, J., Gao, X., Li, Z.: Facial attribute capsules for noise face super resolution (2020)
31. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: a face detection benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

32. Yu, X., Fernando, B., Hartley, R., Porikli, F.: Super-resolving very low-resolution face images with supplementary attributes. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 908–917 (2018)
33. Yu, X., Porikli, F.: Hallucinating very low-resolution unaligned and noisy face images by transformative discriminative autoencoders. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5367–5375 (2017)
34. Yu, X., Fernando, B., Ghanem, B., Porikli, F., Hartley, R.: Face super-resolution guided by facial component heatmaps. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11213, pp. 219–235. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01240-3_14
35. Yuan, Y., Liu, S., Zhang, J., Zhang, Y., Dong, C., Lin, L.: Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 814–81409 (2018)
36. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. CoRR abs/1807.02758 (2018). http://arxiv.org/abs/1807.02758
37. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. CoRR abs/1703.10593 (2017). http://arxiv.org/abs/1703.10593